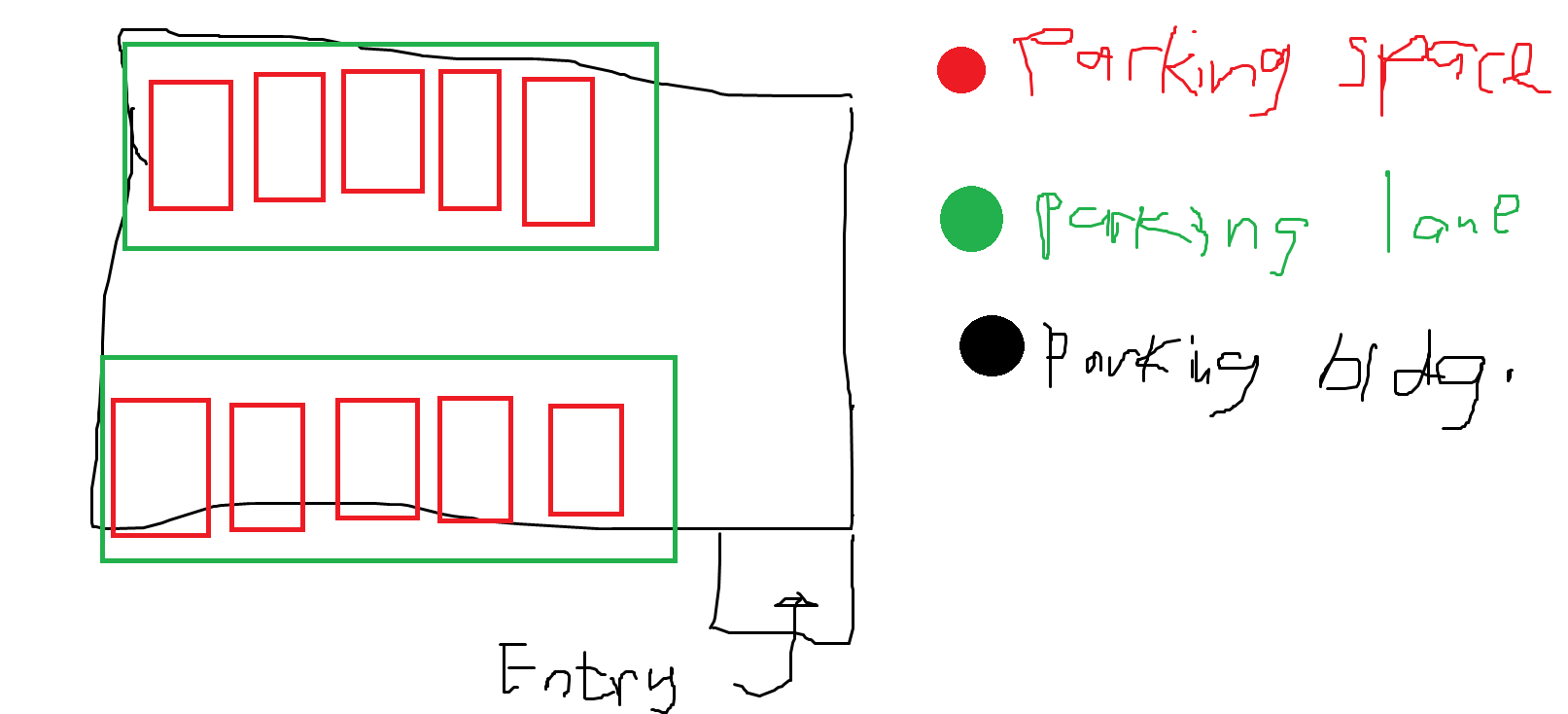
**Day 1 (dec 20)**

**Rescoping project**

As stated in the project proposal, there are some of the things I might need to further clarify and take out. Here are the features that the program will retain:

* A parking space (1 stack) will contain the vehicle (a random string for the plate number)
* A parking lane (1 list) will contain 5 parking spaces
* A parking building (one 2D list) will contain 2 parking lanes  
  *this illustration might help:*
* The graphics is CLI based: standard output and text-based input.

I decided that it’s better if the interaction between the program and the operator (user) models the real world. So, this goes like this:

* the queue of cars outside indefinitely grows (this growth and the plate number is handled with a random number generator);
* the list of stacks (array of parking lots) randomly shrinks;
* both the time of the shrink (stack) and growth (queue) is controlled by the user for the purposes of simulation.

After those are implemented, the features I want to expand more are:

* have the user be able to click and drag cars and collide with any objects present in the parking bldg.
* If time allows, the graphics will resemble the above picture using raylib

Day 2 (dec 21):

**Base project data structures**

* Making a pseudocode
  + I found it hard to wrap how understand the requirement. The pseudocode below shows my initial attempt. However, I realized that in python, in order to enforce such a structure of data, we can use classes.

Que = []  
  
// when queue insert  
When lane a is not empty  
 Insert from queue to whatever stacks that are empty  
Or when lane b is not empty  
 Insert from queue to whatever stacks that are empty  
  
LaneA = [Stack(), stack(), stack(), stack()]  
LaneB = [Stack(), stack(), stack(), stack()]  
  
PrkingSpace = [LaneA, LaneB]

* I made a class for each said thing (parking space, lane, and bldg.) for a more coherent way to interact.
* I also made a few changes from the lab activities we did on stacks and queues. I also did some tests to check whether it was okay for this project. Although I am not sure how will I handle the errors. I asked chatgpt for some guidance, particularly for whatever this means: OverflowError, and IndexError. I might have to do a proper unit test, but im feeling not that too inquisitive this week.

**Code structuring**

* I was wondering how I could structure the codebase the same way we do in c/cpp. These references [1; 2] has a weird work around and I don’t want to work with abstract base class (abc) package (*python is weird)*.
* I initially decided I will store all the required data structures in one file. However, after reading further on how python projects are structured [3; 4], there was a git repository that provides a sample python codebase [5]. I cant really accommodate this kind of structure fully due to the constraints of the project, and I understand little craps. Where dahek is the main function? ?? Why do I need to run a python program the same way I run scripts and why is the main functions only included in unit tests?? ??? ? ? I thought I understood python that much, but I realized I understood less than 10% of the full experience. Sorry for the rant. Looking a bit further, I found that I can use pyinstaller to build an executable [6]. And I seem to have a bit of trouble setting it up as well as running the test cases from the structure provided in samplemod [5].

Day 3 (dec 24)

**Actually working with the base case**

* I am feeling it that I’m stretching this project way too much, so I’ll just work with the base cases and worry about the code structure later. I have made a github repository for tracking this (see appendix).
* From the three classes I have made (parking space, lane, and bldg), I have provided these methods:

Park space:

* Add vehicle
* Pop vehicle
* Get vehicle
* Is empty (private)
* Is full (private)

And starting with the parking lane, it is requiring quite a bit of brain power. I am thinking that I need to handle the randomized sa pagalis nung mga vehicle in this class. I would first need to randomize which in the list (parking space) will get out, then have a time multiplier that controls the whole thing. To get such effect, I need time generator that I got from Code Pope [7].

*However, I tried sampling if we are to print the lane park without contents as I am using the \_\_repr\_\_ to print the first element of the stack. I wanted to see a list of None when printing the lane, however I only see a list of memory locations. The second commit shows that. I don’t think stack is appropriate for this case.*

*Well that was a joke. I just realized I don’t need to access the first element of the stack. In the \_\_repr\_\_ I just need to check whether if the stack is empty or not as well as some other problems. The third commit shows the fix that I can still use the stack as dsa*

Going back to the parking lane implemention of parking lane, I cannot seem to iterate over the parking lane object, its length, get the content, and remove the vehicle. Omy snarky python. I thought this was automatic. I added the iter, len, and getitem special methods thanks to chatgpt hehe. The fourth commit addresses such. I also separated this test case to the unitTests.py inside the unit test folder (now named as unitTests) as I wanted a clean main function. I just realized that the context.py isn’t that much needed when it’s this small and I cant exactly resolve how to use this dependency. I just left two choices that you may need to comment whatever whether I want to *standalonely* compile the unit test, or just call the tests in the main function. But dang, I feel so dumb now with all these dependencies.

**Reference:**

1. <https://stackoverflow.com/questions/70382261/define-function-and-classes-separately-from-their-declarations>
2. <https://stackoverflow.com/questions/12542111/separating-class-definition-and-implementation-in-python>
3. <https://realpython.com/python-application-layouts/>
4. <https://stackoverflow.com/questions/448271/what-is-init-py-for>
5. <https://github.com/navdeep-G/samplemod>
6. <https://stackoverflow.com/questions/51455765/how-to-build-multiple-py-files-into-a-single-executable-file-using-pyinstaller>
7. <https://stackoverflow.com/questions/61713551/how-to-properly-use-time-time>

**Appendix**

Github repository: <https://github.com/Jeo0/perking-spaiyc>